

一种改进的关联规则挖掘算法在高校招生录取中的应用研究

成平广, 韦鹏程

(重庆教育学院 计算机与现代教育技术系, 重庆 400067)

摘 要: 详细分析了传统关联规则 Apriori 算法的不足, 提出了一种改进的关联规则快速挖掘算法. 该算法具有快速去掉冗余规则和不需查找频繁项等优点, 适用于海量数据库的关联规则挖掘. 针对当前高校招生录取后大量考生流失问题, 使用该算法对某地区考生信息进行数理分析和仿真实验, 挖掘了隐含的有用信息, 为高校招生录取提供决策性的作用.

关键词: 数据挖掘; 关联规则; 招生录取

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2010)05-0189-04

Research on an Improved Association Rules Data Mining Algorithm and its Application in College Enrolling Affairs

CHENG Ping-guang, WEI Peng-cheng

(Department of Computer and Modern Education Technology, Chongqing Education College, Chongqing 400067, China)

Abstract: Based on analyzing the traditional association rules algorithm, an improved association rules Data Mining Algorithm is presented in this paper. The advantages of presented algorithm are fast removing redundant rules and not by the frequent items, so it is suitable for the massive data mining based on association rules. In view of the current large number of college students admitted after the loss of the candidates, some college enrollment data was mined based on the proposed algorithm. The results show that the proposed algorithm can mine useful information and play an important role in college enrolling affairs.

Key words: data mining; association rules; college enrollment

1 引言

数据挖掘(Data Mining)技术产生十几年来得到了较为广泛的应用,但在高等教育领域,尤其在高考工作中的应用目前处于起步阶段.文中根据传统关联规则 Apriori 算法的不足,提出了一种改进的关联规则快速挖掘算法,并使用该算法对某地区考生信息进行数理分析和仿真实验,挖掘了隐含的有用信息,为高校招生录取提供决策性的作用,具有重要的实用价值和广阔的发展前景.

2 关联规则

2.1 关联规则的思想

关联规则的一般性描述是^[1]: 设 $I = \{i_1, i_2, \dots, i_m\}$ 是 m 个不同项目的集合, D 是针对 I 的事物集合,每一事物包含若干个项目 $i_1, i_2, \dots, i_k, \dots$ 属于 I , 一个关联规则是一种蕴涵: $X \rightarrow Y$, 其中, $X \subset I$, $Y \subset I$, 并且 $X \cap Y = \Phi$, X 称作规则的前提, Y 是结果. 对于关联规则 $X \rightarrow Y$ 成立的条件是:

① 它具有支持度 S , 即事务数据库 D 中至少有 $S\%$ 的事务同时包含 X 和 Y ;

收稿日期: 2009-04-15; 修回日期: 2009-07-24

基金项目: 国家自然科学基金项目(60703035); 重庆市科委自然科学基金计划项目(CSTC2006BB2254); 重庆市教委项目(KJ061501, KJ071504)

② 它具有置信度 C , 即在事务数据库 D 中包含 X 的事务至少有 $C\%$ 同时也包含 Y .

关联规则挖掘问题可以分为以下两个问题:

① 找出所有满足最小支持度 $\min_sup$ 的项集, 称为频繁项集(或者大项集);

② 根据频繁项集, 产生所有大于最小置信度的规则. 设 $ABCD$ 是频繁项集, 如:

$$\text{support}(AB)/\text{support}(A) \geq \min_conf,$$

则 $A \rightarrow B$ 成立.

由于第 ② 步相对而言比较容易, 因此目前的研究重点在第 ① 步, 即找出频繁项集.

2.2 Apriori 算法的描述

最经典的关联规则挖掘算法是 Apriori 算法^[2], 其思想是利用已知的高频数据项集推导其他高频数据项集. Apriori 算法是一种宽度优先算法, 算法步骤如下:

① 在第一次扫描中, Apriori 算法计算 D 中所有单个项目的支持度, 生成所有长度为 1 的 1-频繁项 L_1 .

② 如果 L_{k-1} 已生成, 现在可用它来生成 L_k , 若有两个 L_{k-1} , 如果其前面的 L_{k-2} 相同, 而最后一项不同, 则将这样的两个 L_{k-1} 进行连接后得到候选 k -项集的集合 C_k .

③ 对候选 k -项集进行剪枝, 从 C_k 中删除所有 $(k-1)$ -子集不全包含在 L_{k-1} 中的项集.

④ 扫描数据库事务 D , 对于其中的每一个事务, 如果它包含 C_k 中的候选项集 c , 则将 c 的计数值加 1 (扫描开始时, 初始值为 0). 扫描 C_k , 计算这些候选项集的支持度, 删除其支持度低于用户给定的最小支持度的项集. 最后, 生成所有长度为 k 的频繁项集 L_k .

⑤ 重复步骤 ② 到 ④, 直到 L_k 为空.

⑥ 对 L_1 到 L_k 取并集即为最终的频繁集 L .

2.3 Apriori 算法存在的不足

Apriori 算法核心思想把发现关联规则的工作也主要分为两步: 通过迭代检索出事务数据库中的频繁项集和从频繁集中构造出满足用户最低信任度的规则. 很显然 Apriori 算法在性能上有两个瓶颈:

(1) 需要多次扫描数据库, 需要很大的 I/O 负载, 对每次循环, 后选集中的每个元素都必须通过扫描数据库一次来验证其是否加入频繁集中.

(2) 产生庞大的后选集, 后选集是以指数形式增长的. 如此大的后选集对时间和主存空间都是一个挑战.

3 一种改进的关联规则挖掘算法

3.1 改进算法的思想

该算法首先构造事务规则树, 将事务规则树合并为规则链, 然后在规则链上构造事务规则树模型, 在构造事务规则树模型的算法完成后, 所提取到的就是所有的项目序列, 再对项目序列进行计算, 就可求出所有的关联规则. 该算法不需要查找频繁项, 直接找出关联规则, 方法快捷灵活, 特别适用于动态的、海量的数据库关联规则挖掘^[3-7].

3.2 改进算法的描述

首先遍历数据库找出有序集 $L = \{i_1, i_2, \dots, i_k\}$ 为满足支持度的频繁 1 项集, 令 $M = \{i_1, i_2, \dots, i_k\}$ 是与 L 一一对应的结点集合.

(1) 开始.

(2) 以 $i_1, i_2, \dots, i_k$ 为根结点分别构造事务规则树 $T_1, T_2, \dots, T_k$. 循环: m 以 1 为步长, 从 1 到 k , 构造事务规则树 T_m .

① 先对 $i_1, i_2, \dots, i_k$ 进行排序: 将项目 i_m 放到有序集 M 的第 1 个位置. 对其余项目按已有先后位置重新排列下标号, 形成新的有序集: $M' = \{i'_1, i'_2, \dots, i'_k\}$;

② 求解规则链第 1 层项目序列集合 L_1 . 令 i'_1 的事务集与 M' 中的其他项目的事务集进行‘交’运算, 将满足支持度的项集形成序列保存到 L_1 中. 其中, 每个项目序列里 i'_1 式为第一元素, 其他项目为第二元素;

③ 循环: t 以 1 为步长, 从 2 到 k , 求解规则链第 2, 3, $\dots$, k 的层项目序列集合 $L_1, L_2, \dots, L_k$; 循环: j 以 1 为步长, 从 1 到 $t-1$, 求解 L_1 , 对 $L_2, L_3, \dots, L_{k-1}$ 进行扩充.

若 $i'_1 (2 \leq t \leq k)$ 是第 t 层的结点, 则 i'_1 依次与 L_j 里的序列进行下列操作:

首先, 令 L_j 中的每个序列对应项集的事务集与 i'_t 的事务集‘交’运算, 若结果满足支持度, 就将该序列与 i'_t 形成新的项目序列放入 L_t , 其中, i'_t 是序列中最后一个元素. 并且后缀是‘交’运算结果集合的长度.

其次, 令 L_i 中的每个序列对应项集的事务集与的事务集‘交’运算, 若结果满足支持度, 就将该序列与 i'_j 形成新的项目序列放入 L_j , 对 L_j 进行了扩充. 其中, i'_j 是序列中最后一个元素, 并且后缀是‘交’运算结果集合的长度.

④ 根据 $L_1, L_2, \dots, L_k$ 里的序列,按照路径规则法求出所有的关联规则放入规则集 P 中。

下面是一条规则链上实现构造事务规则树模型的算法:

其中 L_t 是第 t 层上的序列集合, L_s 是 L_t 中第 s 个项目序列, Q 是冗余序列集合,最终得到的项目序列放到序列集合 PK 中。

```
(1)  $s = 1$ 
(2) for ( $k = 2; k \leq n; k++$ ) do // 求出规则链第一层的序列集合  $L_1$ .
(3) if ( $x_k = |RT_{i1} \cap RT_{ik}| \geq \min\_sup$ )
(4)  $\{l_s = \langle i_1 : y_1; i_k : x_k \rangle$ 
(5)  $L_1 = \bigcup l_s$ 
(6)  $s++$ ; }
(7) for ( $k = 2; k \leq n; k++$ ) do // 求出  $L_2, L_3, \dots, L_n$ ,
判断  $i_k$  与  $L_j$  里的序列是否能形成新的序列,如果形成新的序列,分别放入  $L_k$  和  $L_j$  中
(8) for ( $j = 1; j \leq k-1; j++$ ) do
(9) {for ( $s = 1; s \leq |L_j|; s++$ ) do
(10)  $\{x_k = 0$ 
(11)  $R = \langle l_s(l_s \in L_j) \cup i_k : x_k \rangle$  // 若  $i_k$  与序列  $l_s$  里的项目形成的项集满足最小支持度. 将  $l_s$  放入冗余序列集合  $Q$ .  $\langle l_s, l_k : x_k \rangle$  放入  $T$  里
(12) if ( $m = |each RT_{im}(i_m \in l_s) \cap RT_{ik}| \geq \min\_sup$ )
(13)  $\{x_k = m$ 
(14)  $Q = \bigcup l_s$ 
(15)  $T = \bigcup R$ 
(16) }
(17) for ( $s = 1; s \leq |L_k|; s++$ )
(18)  $\{x_j = 0$ 
(19)  $R = \langle l_s(l_s \in L_k) \cup i_j : x_j \rangle$  // 若  $i_j$  与序列  $l_s$  里的项目形成的项集满足最小支持度. 将  $l_s$  放入冗余序列集合  $Q$ .  $\langle l_s, i_j : x_j \rangle$  放入  $L_j$ 
(20) if ( $m = |each RT_{im}(i_m \in l_s) \cap RT_{ij}| \geq \min\_sup$ )
(21)  $\{x_j = m$ 
(22)  $Q = \bigcup l_s$ 
(23)  $L_j = \bigcup R$ 
(24) }
(25)  $L_k = \bigcup T$ 
(26) }
(27)  $P_h = L_1 \cup L_2 \cup L_3 \cup L_n$ ; // 将所有的序列放入以  $i_h$  为根结点的规则链的集合  $P_h$ 
(28)  $P_h = P_h - Q$  // 消除冗余序列
将  $n$  条规则链求出的项目序列放入集合  $P$  中.  $P = P_1 \cup P_2 \cup \dots \cup P_n$ . 对  $P$  中的序列运用路径规则法和规则过滤求出无冗余的关联规则.
(3) 对所求出的事务规则树  $T_1, T_2, \dots, T_k$  的规则进行规则过滤以消除冗余.
```

(4) 结束.

该算法根据最大频繁集长度 n 需要构造 n 条规则链,假设最大频繁集中的项目位置固定,当以某一个项目 i_k 为根结点建立 1 条规则链时,将 i_k 放到最大频繁集的第一个位置,对其余项目按已有先后位置重新排列,每个位置的项目对应规则链中相应层次的结点。

3.3 算法的性能分析

通过上述算法的描述和分析,可以得出该算法主要具有如下优点:

(1) 算法可由所给事务数据库直接查找出所有的无冗余的关联规则,省略了其中查找频繁项集的步骤。

(2) 算法的时间复杂度是 $O(|L_x| \cdot k^3)$, k 是最大频繁集长度, L_x 是规则链中第 x 层的项集集合,规则链是构造事务规则树模型的数据结构.传统的规则生成方法的复杂度是 $O(N \cdot 2^k)$,随 k 的增长呈指数倍增长。

该算法可以观察所求出的关联规则,直接得到无冗余的频繁项集,也可以用于求解频繁项集.所以该算法较以前的一些关联规则挖掘算法有所改进,是高效、可行的关联规则挖掘算法。

4 在高校招生录取中的应用研究

实验数据来自某地区 2005 年的考生信息库,记录总数为 81653 条.考察录取为普通专科学学生的志愿信息、成绩信息和综合素质等对其录取后是否入学报到的关联挖掘.扫描数据库,统计学生信息及可能入学报到的支持度计数信息如表 1 所示。

表 1 考生部分信息表

挖掘类型	人数	报到人数	事务标识
第一志愿	10604	9249	A1
第二三志愿	638	278	A2
调剂志愿	411	122	A3
高分考生	3146	2364	B1
上线考生	8507	7285	B2
特长考生	693	148	C1
普通考生	10960	9501	C2

记第一志愿为事务 A1,第二三志愿为事务 A2,调剂志愿为事务 A3;高分考生为事务 B1,上线考生为事务 B2;特长考生为事务 C1,普通考生为事务 C2;入学报到为事务 D. 则计算出如下关联规则:

(1) $\text{support}(A1 \Rightarrow D) = P(A1 \cup D) = 9249 / 81653 = 11.32\% = \text{support}(D \Rightarrow A1);$

$\text{confidence}(A1 \Rightarrow D) = P(D \cup A1) = 9249/10604 = 87.22\%$;

$\text{correlation}(A1, D) = P(A1 \cup D)/P(A1)P(D)) = 18.56 > 1$.

据此可得如下形式的关联规则:

$A1 \Rightarrow D$ [support = 11.32%, confidence = 87.22%].

(2) $\text{support}(B2 \Rightarrow D) = P(B2 \cup D) = 7285/81653 = 8.92\% = \text{support}(D \Rightarrow B2)$;

$\text{confidence} = (B2 \Rightarrow D) = P(D \cup B2) = 7285/8507 = 85.63\%$;

$\text{correlation}(B2, D) = P(B2 \cup D)/P(B2)P(D)) = 24.55 > 1$.

据此可得如下形式的关联规则:

$B2 \Rightarrow D$ [support = 8.92%, confidence = 85.63%].

同理可得其他形式的关联规则. 部分关联规则如表 2 所示.

表 2 部分关联规则表

关联规则	支持度(%)	置信度(%)	相关度
$A_1 \Rightarrow D$	11.32	87.22	>1
$A_2 \Rightarrow D$	0.34	43.57	>1
$A_3 \Rightarrow D$	0.15	29.68	>1
$B_1 \Rightarrow D$	2.90	75.14	>1
$B_2 \Rightarrow D$	8.92	85.63	>1
$C_1 \Rightarrow D$	0.18	21.35	>1
$C_2 \Rightarrow D$	11.64	86.68	>1

以上规则说明如下:

(1) 被录取考生中有 11.32% 的同学填报专科一阶段的第一志愿并报到, 填报第一志愿的考生有 87.22% 的可能会报到, 报到的考生有 95.85% 的可能会填报第一志愿;

(2) 被录取考生中有 8.92% 的同学不是高分考生(即分数没有达到本科控制分数线)并报到, 上线考生(即非高分考生)有 85.63% 的可能会报到, 报到的考生有 75.5% 的可能是上线考生;

(3) 被录取考生中有 11.64% 的同学不是特长考生(有特长并获得省部级奖励的)并报到, 普通考生(即非特长考生)有 86.68% 的可能会报到, 报到的考生有 98.46% 的可能会是普通考生等.

5 结束语

文中在研究和分析传统关联规则 Apriori 算法不足的基础上提出了一种改进的关联规则快速挖掘算法, 并针对我国当前高校招生录取后考生大量流失现象将其应用到高考信息中进行数据挖掘和知识发现, 快速提取出大量隐含的有价值的信息为高校招生录取做出决策参考, 具有一定的实用性和有效性.

参考文献:

[1] Hand D, Maunila H. 数据挖掘原理[M]. 张银奎, 译. 北京: 机械工业出版社, 2003.

[2] Agrawal R, Imidinski T, Swami A. Mining association rules between sets of items in large databases[C]//Proceedings of the ACM SIGMOD Conference on Management of date. San Jose, 1993:207-216.

[3] Min Song, Il - Yeol Song, Xiaohua Hu. Semantic query expansion combining association rules with ontologies and information retrieval techniques[C]// DaWaK. University Heights Newark, 2005:326-335.

[4] 丁卫平, 祁恒, 董建成, 等. 基于关联规则的电子病历挖掘算法研究与应用[J]. 微电子学与计算机, 2007, 24(3):69-73.

[5] 秦亮曦, 李谦, 史忠植. 基于排序 FP 一树的频繁模式高效挖掘算法[J]. 计算机科学, 2005, 32(4):31-33.

[6] 周霆, 张伟, 张泽洪. 基于关联规则的映射聚类算法[J]. 微电子学与计算机, 2006, 23(3):26-29.

[7] 芦洁, 刘志镜. 挖掘关联规则中对 Apriori 算法的一个改进[J]. 微电子学与计算机, 2006, 23(2):10-12.

作者简介:

成平广 男, (1975-), 硕士, 讲师. 研究方向为数据挖掘、计算智能等.

韦鹏程 男, (1975-), 博士, 副教授. 研究方向为信息安全技术、计算智能.